

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: DESIGN-TIME CREATION OF RUN-TIME MODULES
THAT USE CATEGORIZATION

APPLICANT: CHRISTOPHER RONNEWINKEL

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV479290563US

April 12, 2004
Date of Deposit

Design-Time Creation of Run-Time Modules that Use

Categorization

TECHNICAL FIELD

This invention relates to application software development, and more particularly to software development involving categorization.

BACKGROUND

5 This document relates to design-time processes and methods of creating executable run-time modules that use categorization schemes. Exemplary applications for such processes and methods relate to enterprise computing systems. One such enterprise computing system involves software for performing business processes that involve responding to large volumes of
10 inbound communications from customers. Such inbound communications are typically in the form of email messages, but they may also include other forms of inbound data, such as information entered by a customer over the Internet using web self-service, web forms, and the like.

 The productivity of an individual agent who is responsible for responding
15 to large volumes of inbound communications affects the cost of performing that particular business process. However, in controlling the cost, the quality of the responses should be maintained so that the customer is well served. Thus, systems that are used to respond to inbound communications should provide high quality responses in an efficient manner.

Business application systems that respond to high volumes of inbound emails may be called Email Response Management Systems (ERMS). If some of the steps in the ERMS process of responding to an inbound email are automated, then the enterprise can realize efficiency and productivity gains. An enterprise can use enterprise computing systems that run ERMS software to automate steps of the ERMS process. As such, an enterprise can use ERMS software to enhance productivity.

To realize maximize productivity gains in the run-time environment, an enterprise usually must first customize the ERMS software in the design-time environment. Specifically, the ERMS software must be configured to perform the particular business process steps required by a particular enterprise. In addition, the enterprise must be able to reconfigure the ERMS software to reflect changes in the enterprise's business process steps over time. Thus, an enterprise should be able to customize and to maintain ERMS software so that maximum productivity gains can be realized.

SUMMARY

A design process for implementing coherent categorization includes defining a categorization scheme, linking the categories to stored information, and providing at least two executable response modules for performing business processes using a single categorization. The design process can be used in the design-time environment to develop coherently categorized run-time modules. Although coherently categorized run-time modules perform different steps in a business process, they use stored information linked to the same category. As

such, coherently categorized run-time modules perform steps in a business process using a single categorization step. This provides various performance and integration advantages to an enterprise, thereby reducing costs and improving efficiency.

5 In one aspect, a method of defining a system for responding to an incoming electronic message (IEM) includes defining a categorization scheme having a plurality of hierarchically organized categories. The method also includes linking stored information to each defined category. In addition, the method includes providing multiple response procedure modules. Each
10 response procedure module, when executed under user control, uses the stored information that is linked to a selected one of the defined categories. The selected category is determined by analyzing the content of the IEM.

 The foregoing aspect may be modified. For example, the stored information may include business objects. Such business objects may be any of
15 quick solutions, experts, or response templates. Furthermore, the defined system may perform a business process, which in some cases may be an ERMS. The system may also be a contact center, or an interaction center.

 In other examples, at least one of the multiple response procedures may perform a step of responding to an IEM. The IEM may be an email, or it may be
20 received via Internet self-service. The categorization scheme may be hierarchical and have at least two levels of categories.

Other examples involve defining queries for each category. These defined queries may be evaluated to analyze the content of the IEM to select categories for which the corresponding defined queries match the content of the IEM.

In another aspect, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, cause a processor to perform operations to define a system for responding to an IEM in accord with the method aspect described above.

The computer program product aspect may be modified such that at least one of the response procedure modules, when executed, performs a step in an ERMS business process, performs a step other than a step in an ERMS business process, or performs a step in an interaction center.

In still another aspect, a computer-implemented system for defining a system for responding to an IEM includes a graphical user interface (GUI). The GUI is capable of defining categorization schemes to have a plurality of hierarchically organized categories, and capable of linking stored information to each category. The system also includes multiple response procedure modules. Each response procedure module, when executed under user control, uses the stored information that is linked to a selected one of the defined categories. The selected category is determined by analyzing the content of the IEM.

The details of one or more implementations are set forth in the accompanying drawings and the description below. These aspects and implementations may provide various advantages. For example, implementations provide enterprise system end users a method by which to develop and to maintain run-

time modules that reap the benefits of coherent categorization. The benefits of coherent categorization include, for example, improved auto-suggestion of responses to incoming requests for information. Accordingly, the methods reduce the enterprise's reliance on software vendors and consultants by enabling the enterprise to employ self-help to create and maintain customized business software, especially business software that involves coherent categorization. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

Fig. 1A is an enterprise computing system.

Fig. 1B is a flow chart of steps for creating executable process modules that use coherent categorization.

Figs. 2A-2B are run-time flow diagrams of a business application using a coherent categorization scheme.

Fig. 3 is a categorization scheme.

Fig. 4 is a portion of the categorization scheme of Fig. 3 with additional detail.

Fig. 5A is a conceptual diagram of the process steps performed when manually selecting a category in a categorization scheme in accordance with Fig. 2B.

Fig. 5B is a user interface for manually selecting categories using the process of Fig. 5A.

Fig. 6 is a block diagram of an ERMS business process using coherent categorization schemes of Fig. 3.

Figs. 7-9 are screen shots of an exemplary run-time graphical user interface (GUI) for the ERMS of Fig. 6.

Fig. 10 is a design-time GUI for defining queries for the categorization scheme of Fig. 3.

Fig. 11 is a flowchart of the steps to define a categorization scheme using the GUI of Fig. 10.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

This document relates to creating modules at design-time that, when executed at run-time, use coherent categorization to perform some business processes. A system that includes the run-time modules and categorization scheme created using this process will yield the benefits of coherent categorization, including the ability to more efficiently process information that is processed using multiple business process steps. As a further advantage, an enterprise that uses coherent categorization is simpler and less costly to maintain. Moreover, a categorization that is well-defined across multiple business processes is an integrated semantic representation of the business processes in an enterprise. As such, business processes can be analyzed based on coherent categorization schemes to provide insights into the structure of an enterprise's business.

Off-line analysis is generally performed on data that is historic, stored in a database, and consolidated. In contrast, on-line analysis is performed at run-time using run-time data that is not necessarily stored in a database. As will be seen from the following description, coherent categorization can enhance both types of systems analysis.

Methods for creating such modules involve processes that may be manually performed or partially automated. In the design-time environment, a developer can use various conventional software development tools to develop the run-time modules. In addition, the coherent categorization schemes may be created at design-time using various techniques, such as by using a graphical user interface (GUI) to define the categorization schemes. The overall process requires that the run-time modules be created at design-time in such a way that more than one run-time module can use the result of a single categorization.

When executed, each run-time module performs a step in a business process. Accordingly, the developer creates run-time modules in such a way that they can use categorization schemes in a coherent manner, i.e. that more than one run-time module can use the result of the single categorization at run-time. When a business process receives an input that is to be processed using a series of business process steps, how the input is categorized determines the action that each of the business process steps will take. Process efficiencies can be gained, thereby improving productivity, if the business processes and categorization schemes are structured so that a single categorization result can be used by more than one of the business process steps. Accordingly, coherent

categorization is achieved, and advantages obtained, when multiple business process steps can directly use the result of a single categorization.

For ease of understanding, the design-time process will first be introduced in the context of the computing environment in which it may be performed. With that introduction, the design-time process steps for creating the run-time system are explained using a flow chart. Next, run-time aspects of using categorization schemes are presented. Then, the details of the run-time coherent categorization scheme are described, followed by a description of an exemplary design-time user interface for creating categorization schemes.

Beginning with the computing environment in which the run-time modules are defined and executed, Fig. 1A shows an enterprise computing system 1 having a design-time environment 2 and a run-time environment 3. In the design-time environment 2, a developer can develop program modules and categorization schemes using various tools commonly available to the developer. The programming tools may be accessed through the terminal 4. The development tools may include, for example, debuggers, compilers, and editing programs that are stored in a tools repository 5. The developer may also use the terminal 4 to create, to store, and to edit and review created business processes in a design-time memory 6.

The design-time environment 2 and the run-time environment 3 may each be connected to a network 7, which may be an intranet. Connected to the network 7 is a stored information repository 8, which may comprise a plurality of databases for storing information. One of these databases is a categorization

scheme database 9 for storing categorization schemes. Another database is a process modules database 10 for storing programs that may be executed at run-time using categorization schemes stored in the categorization scheme database 9. A third database is a business objects database for storing business objects that may be associated with individual categories of the categorization schemes in the categorization schemes database 9. Information in each of these databases can be accessed from both environments 2, 3. In general, the information in these databases is defined by a developer in the design-time environment 2, and used to perform business processes by a user in the run-time environment 3.

A user in the design-time environment 2 may use a computer terminal 4 to enter, modify, and remove information that may include information stored on the stored information repository 8. Similarly, a user in the run-time environment 3 uses a computer terminal 12 to perform run-time applications that can use access, modify, and delete information stored in the stored information repository 8.

A user in the run-time environment 3 can use a terminal 12 to gain access to the network 7 and to the information stored in the stored information repository 8. In particular, the run-time user may load executable program instructions stored in the process modules database 10, and load data stored in the categorization schemes and business objects databases 9, 11, into a memory 13. While performing various business processes, portions of the loaded instructions

are executed on a processor (not shown) under the control of the user via the terminal 12.

Referring again to the design-time environment 2, software developers, for example, use various tools, including editors, debuggers and compilers, in order to develop software modules, user interfaces, executable programs, and the like, for use in the run-time environment 3. In developing such run-time environment applications, a user in the design-time environment 2 loads stored information from the stored information repository 8 through the network 7 and into the terminal 4 in order to manipulate that information. For example, the design-time environment 2 developer may load application programs from the stored information repository 8 and use those application programs to create, for example, categorization schemes. These created categorization schemes can incorporate business objects and other data that is also loaded from the stored information repository 8 into the memory 6 and displayed on the terminal 4. The developer can then store the modified categorization scheme, of this example, back into the stored information repository 8 where it may be accessed from the terminal 12 in the run-time environment 3. Accordingly, the user in the design-time environment 2 can store data and programs in the stored information repository 8 that the user in the run-time environment 3 can use to perform run-time applications. Moreover, the run-time environment user may also manipulate stored information in the stored information repository 8. As such, the run-time environment 3 may affect the data in the stored information repository 8 that is subsequently used within the design-time environment 2.

The enterprise computing system 1 may be connected to additional networks, for example, the Internet. Although not shown in Fig. 1A, the Internet may be connected to the design-time environment 2, the run-time environment 3, or the network 7 using standard communication interface hardware and software techniques.

The method steps that a developer could use to create run-time modules that use coherent categorization are illustrated in the flowchart 18 of Fig. 1B.

After starting at 19, the developer collects at 20 any relevant data from the stored information repository 8. Specifically, the developer may load all relevant

information from any of the databases 9, 10, 11 over the network 7, and into the memory 6. As will be discussed in detail with reference to Figs. 10-11, the

developer next defines a categorization scheme at 21. At 22, the developer next links categories from the defined categorization scheme to business objects in the stored information repository 8. At 23, the developer develops a first program

that, when executed, performs a first step in the business process using business objects linked to a selected category. At 24, the developer develops a second

program that, when executed, performs a second step in the business process using business objects linked to the selected category. At 25, the developer

compiles the code for the first and second programs into process modules in an executable format. Finally at 26, the developer stores the categorization

schemes and process modules in the stored information repository 8 for execution in response to user input at run-time. The process of creating run-time modules that use categorization is completed at 27.

Accordingly, a developer can create categorization schemes and run-time modules by following the method steps of the flow chart 18. With reference to Figure 1A, a developer in the design-time environment 2 can load data from the databases 9, 10, 11 into the memory 6. The developer can define a categorization scheme that refers to this data using software development tools from the tool repository 5. Similarly, the developer can use software development tools to develop and maintain executable process modules that use the defined categorization schemes to perform business processes. In addition to defining categorization schemes and process modules, the developer can add, modify, or delete data that may be stored in the business objects database 11. Then, the developer can store the categorization schemes, process modules, and business objects into the databases 9, 10, 11, respectively, for use in the run-time environment 3.

The design-time process steps of Fig. 1B may be performed to create run-time modules that perform business processes, such as the business process steps shown in Figs. 2A-2B. The exemplary business process steps in Fig. 2A may relate to a business application, for example, an ERMS. After a business application has been designed in the design-time environment 2 according to the process steps of Fig. 1B, the business application can be executed in the run-time environment 3.

Coherent Categorization

After a business application has been designed in the design-time environment 2, the business application can be executed in the run-time

environment 3. Referring to Figs. 2A-2B, a run-time flow diagram illustrates execution of a business application that uses coherent categorization to perform multiple business process steps. Figs. 2A-2B illustrate use of coherent categorization schemes in two exemplary versions of the business application, namely a manual categorization, and an automated categorization.

In the manual version shown in Fig. 2A, the business application 28 responds to an input signal 30 by producing an output signal 32. A first business process module 34 executes instructions to perform one of the business application 28 business process steps. To perform that step, the module 34 categorizes the input signal 30. As will be shown with respect to Fig. 5B, a human user manually selects a category displayed on a user interface. The business process module 34 displays categories from a categorization scheme 36, and may limit the displayed categories to those that are relevant to performing the business process step 34. Moreover, because the business application is structured to use the categorization scheme 36, the selected category 38 may be used by a subsequent business process module 40 in the presence of the input signal 30.

Linked to the selected category 38 are linked business objects (BO's) 44. The linked BO's 44 may include information, such as, for example, experts 46, quick solutions 48, and response templates 50. Collectively, these linked BO's 44 may be used in multiple business processes, including business processes 34, 40. Where, for example, the business application is an ERMS, the input 30 may be an incoming e-mail message from a customer. In that case, the business

application 28 is used to respond to incoming e-mail messages by providing, for example, a reply e-mail message as the output signal 32.

The categorization scheme 36 is used in a coherent manner because it identifies a selected category 38 that provides relevant BO's 44 to more than one business process module, namely modules 34, 40. As such, the same categorization of an input signal is used to perform more than one business process step. In other words, each business process does not perform its own categorization, as was the case in prior art systems.

A categorization is performed in response to a particular input signal 30, and business processes are performed in reaction to (or in the presence of) a particular input signal. As such, the particular selected category 38 is relevant to a business process only with respect to the content of a particular input signal 30. The selected category may be different for each new incoming input signal. Accordingly, the particular category that is selected within the categorization scheme 36 will depend upon the content of a particular input signal 30.

In the automated version of this example shown in Fig. 2B, a categorization scheme is used to automatically suggest a category, but a user can override that suggestion by manually selecting a different category. For example, the business process module 34 uses the categorization scheme 36 to automatically propose suggested category 39 to subsequently-performed business process module 40. In this example, the business process module 34 may include a content analysis engine for analyzing the content of the input signal 30. In the process of performing the business process module 34, the

human user can choose to use the suggested category 39 to perform the business process module 40, or the user can manually select a different category. Whichever category the user selects may be referred to as the selected category 38. This selected category 38 is proposed to the subsequent business process step 42. As such, business process module 42 may use the selected category 38, or the user can override that choice and select a different category.

To promote the efficient performance of the business process modules 40, 42, linked business objects 44 may be filtered to provide only those business objects that are relevant to the business processes 40, 42. A link 51 represents the link from the selected category 38 to the entire set of linked BO's 44. A link 52 provides a subset of the linked BO's 44 to the business process 40. A link 54 provides another subset of the linked BO's 44 to the second business process 42. The business processes 40, 42 can each use the independent subsets of linked BO's 44 provided by respective links 52, 54 to perform their respective business processes within the business application 28.

The links 52, 54 may provide the same subset of linked BO's 44 to both business processes 40, 42. On the other hand, the links 52, 54 may provide subsets of linked BO's 44 that are different. In the latter case, each of the links 52, 54 may be configured to provide BO's that are of a certain type. For example, if the business process 40 is performed to provide a standard e-mail response template, then the link 52 may be configured to provide only BO's that are of the response template 50 type. Similarly, if the business process 42 is

performed to select documents for attachment to a reply e-mail, then the link 54 may be configured to provide only BO's that are of the quick solutions 48 type, or of the experts 46 type.

Just as business objects are characterized by their type, each link
5 between a category and a business object is characterized by a type. For example, a link from a category to a document of quick solution 48 type may be characterized as being of "is_solution" type. Similarly, links from categories to experts 46 and to response templates 50 may be characterized as being of "is_expert" and "is_response_template" types, respectively.

10 The use of filtering may be illustrated, at least in part, in the context of an exemplary process of responding to an incoming email request that relates to printers. Initially, business process module 34 performs a content analysis of the input signal 30, and it identifies the key word "printer" in the email. The categorization scheme has a category that corresponds to "printer," so that
15 category becomes the suggested category 39. When the business process module 40 is performed, the suggested category 39 may be displayed to the user as a proposed category. If the user determines that the email relates specifically to "laser printers," the user can override the suggested category by manually selecting a "laser printers" category that is a child of the "printers" category.
20 Accordingly, the "laser printers" category becomes the selected category 38. Having selected a category, business objects linked to the "laser printers" category may be provided to the business process module 40 by the link 52. If, for example, the link 52 is configured to filter out business objects that are not of

the response template 50 type, then only response templates 50 that are linked to the "laser printers" category are used.

As will be shown in detail below, one exemplary run-time implementation of the coherent categorization scheme automatically displays only those
5 business objects that are in the set of linked BO's 44 associated with the selected category 38. In the exemplary business application 28, BO's 44 that are not relevant to the business process being performed are, at least initially, filtered out. As such, they are not automatically provided to the business process. Instead, only those BO's most likely to be used are initially displayed to the user.
10 Nevertheless, the user can choose to display business objects that are not linked to the selected category, if that is desired.

By initially displaying only the linked BO's 44, and by filtering out linked BO's that are not of the most relevant type to a business process, a categorization scheme reduces the time and effort the agent must spend
15 performing a business process step. As such, the agent can realize productivity and efficiency improvements. If the categorization is also coherent, a single categorization can serve more than one business process step. By reducing the number of categorizations required to perform a number of business process steps, a coherently categorized system further reduces or eliminates
20 unnecessary time and effort the agent must spend to perform multiple business process steps. As such, coherently categorized systems can yield further efficiency and productivity gains over systems that are not coherently categorized.

The selection of categories to perform the foregoing exemplary business process steps depends on the structural details of the categorization scheme itself. The structures of two exemplary categorization schemes that may be used in the business application 28 of Figs. 2A-2B are illustrated in Figs. 3-4. In
5 general, Figs. 3-4 illustrate how categorization schemes can be used to relate business process steps to relevant business objects, as well as how categorization schemes define relationships between categories.

Referring to Fig. 3, a set of business process steps 100 may be performed, either automatically or in response to user input, during the run-time
10 execution of a business application. The steps in the set of business process steps 100 are linked to a set of categorization schemes 105. Each categorization scheme in the set of categorization schemes 105 is linked, directly or indirectly, to multiple categories 110. The categories may be distributed across any number of levels. For example, the categories may be arranged in a hierarchical structure
15 having several levels, or they may be arranged in a flat structure in a single level. In hierarchically structured categories, each category below a top level is linked to one parent in the next higher level, and may be linked to any number of child categories in the next lower level. Parent/child categories may also be referred to as categories/sub-categories. Any of the categories 110 may be linked to one
20 or more business objects 115.

Accordingly, the categorization schemes 105 relate business objects 115 to the business process steps 100. By defining these associations, categorization schemes reflect relationships between business processes and

resources (i.e., business objects), especially stored information, in the enterprise computing system 1. Moreover, if a categorization scheme 105 identifies a selected category from among the categories 110 that subsequently provides relevant BO's 115 to more than one business process step 100, then that categorization scheme 105 may be referred to as a "coherent" categorization scheme. In a business application that includes coherent categorization, a single categorization may be used to provide business objects to multiple business process steps within the business application. As such, the categorization schemes 105 may reflect relationships across multiple business processes.

For example, Fig. 3 shows an interaction record business process step 120 and an ERMS business process step 125. The interaction record business process step 120 is linked by a link 130 to an interaction reason categorization scheme 135. The ERMS business process step 125 is linked by a link 145 to the interaction reason categorization scheme 135, and it is linked by a link 150 to the product categorization scheme 140. Each of the categorization schemes 125 and 140 are linked to a number of categories. The interaction reason categorization scheme 135 is shown as having a hierarchical structure, while the product categorization scheme 140 is shown as having a flat structure. Under the interaction reason categorization scheme 135, there is a link 155 to a LEGOLAND® category 160, a link 165 to a Lego® club category 170, and a link 175 to a Lego® products category 180. The categories 160, 180 have further sub-categories. The LEGOLAND® category 160 has a link 185 to an entry fee category 190, a link 195 to an events category 200, and a link 205 to a driving

directions category 210. Similarly, the Lego® products category 180 has a link 215 to a building instructions category 220. Other links and categories may be added or removed from the interaction reason categorization scheme 135 to provide different responses for the business process steps 120, 125.

5 By way of example, each of the categories 200, 210 and 220 is linked to relevant business objects within the business objects 115. For example, the events category 200 has a link 225 to a set of business objects 230. As will be described with reference to Fig. 4, the link 225 represents a set of links, whereby each business object in the set of business objects 230 has a uniquely defined
10 link between each business object in the events category 200. Similarly, the driving directions category 210 has a link 235 to a set of business objects 240, and the building instructions category 220 has a link 245 to a set of business objects 250. The sets of business objects 230, 240, 250 each include experts 46, quick solutions 48, and response templates 50.

15 As has been previously suggested, the sets 230, 240, 250 of business objects are selected from available business objects as being relevant to the categories to which they are linked. As such, the number of business objects of a particular type that are included within the particular set of business objects linked to a category can vary based on the number of business objects that are
20 available. For example, the number of experts that are included in the set of linked business objects 230, 240, 250 depends upon the availability of subject matter experts who have knowledge relevant to the appropriate category. Similarly, the numbers of quick solutions 48 and response templates 50 that are

included in a set of linked business objects 230, 240, 250, depend upon the stored contents of, for example, a knowledge base within the stored information repository 8 (Fig. 1A).

Accordingly, if the interaction record business step 120 is being performed in the presence of an input signal 30 (not shown), then content of the input signal 30 will determine how the categorization scheme 135 is navigated. If the content of the input signal 30 relates to driving directions to LEGOLAND®, then the categorization scheme would be navigated through the link 155 to the LEGOLAND® category 160, and through the link 205 to the driving directions category 210. If the ERMS business process step 125 is subsequently performed while responding to the same input signal 30, then the business process step 125 will automatically receive business objects that relate to the chosen driving directions category 210 from the set of business objects 240.

Thus, in the foregoing example, the performance of the interaction record business process step 120 categorizes the input signal 30 to select and use the driving directions category 210. The selected category may subsequently be used by a later business process step, in this example, the ERMS process step 125. Accordingly, the exemplary categorization scheme just described exhibits coherency because a selected category identified in one step of a business process can be used to perform a subsequent business process step.

Although the Fig. 3 represents only business objects being linked to categories that exist at a lowest level (children) categories in the hierarchy, business objects may be also be linked to any category that is a parent category.

As such, a categorization scheme may be defined such that any category that is selected may be linked to a set of business objects 44.

Additional structural detail of a categorization scheme in accordance with the categorization schemes of Fig. 3 is shown in Fig. 4. In one example, Fig. 4 illustrates the selected category 38 in a magnified portion of a hierarchical categorization scheme 300. The selected category 38 is linked by a link 405 to a parent category (not shown) above it. The selected category 38 is also linked to the linked business objects 44. The selected category may exist at any level in the hierarchical categorization scheme 300. Each of the linked business objects 44 are selected from among all available business objects that are stored, for example, in a database (not shown) in the enterprise computing system 1. The linked business objects 44 may include experts 46, quick solutions 48, and/or response templates 50.

Each of the linked business objects 44 is linked to the selected category 38 by a unique link. Individual experts 46a, 46b, and 46c are linked to the selected category 38 by links 47a, 47b, and 47c, respectively, of the "is_expert" type. Individual quick solutions 48a, 48b are linked to the selected category 38 by links 49a, 49b, respectively, of the "is_solution" type. Individual response templates 50a, 50b, and 50c are linked to the selected category 38 by links 51a, 51b, and 51c, respectively, of the "is_response_template" type. Accordingly, one way to modify the categorization scheme is to modify the links 47, 49, or 51.

Use of the categorization schemes of Fig. 3 in, for example, the manually performed category selection in the business application 28 (Fig. 2B) involves the

identification of one or more appropriate categories from within the categorization scheme 36 (Figs. 2A-2B). An exemplary process for manually identifying a selected category 38 is conceptually illustrated in Fig. 5A. An exemplary user interface suitable for manually selecting a category is presented in Fig. 5B.

5 In the example shown in Fig. 5A (and with reference to Fig. 2B), the contents of the incoming message 30 is analyzed at 520 by, for example, a content analysis engine that searches the message for key words that match queries defined for the categories in the categorization scheme 36. In various embodiments, the key word search can be performed either by a human user, or
10 by a programmed computer. Attributes of a category include properties assigned at design-time, and a category's attributes determine whether a category matches the analyzed content of an incoming message. Content analysis may be performed on the textual content of an unstructured incoming email, for example, by performing query-based categorization, example-based
15 classification (using, e.g., either a nearest neighbor, or a support vector machine algorithm on stored previous examples), a combination of both, or other effective method of content analysis.

 The content analysis engine 520 uses a categorization scheme 36 to automatically suggest an initial category 525. This initial category suggestion
20 becomes the current category 530. If a category is found not to have attributes that match the analyzed content, then no category may be initially suggested.

 With the current category 530 initially determined, an interactive auto-suggest cycle begins. Business objects 535 that are linked to the current

category 530 may be displayed and suggested to the user, thereby allowing the user to quickly access those business objects that are likely to be relevant to the business process step being performed. The user interface, in this example, displays the selected category to the user at 540.

5 With reference to the user's option to choose a different category 38 than the suggested category 39 in Fig. 2B, the user can iterate as needed by selecting a different current category 530. In turn, the user interface will again display the objects 535 that are linked to the updated current category 530. In this manner, the user can choose to use the business objects that are linked either to the
10 initially suggested category 525, or to a manually selected category.

The foregoing manual selection process can be implemented in a graphical user interface that may be accessed while performing a business process step that involves categorization. An exemplary user interface 550 is illustrated in Fig. 5B. The user interface 550 includes a number of fields that
15 contain drop down list boxes (DDLBs). The user can select a category, for example, from a DDLB 555 labeled "Classification 1." When selected, the DDLB 555 will display categories that are in a top-level a hierarchical categorization scheme. With reference to Fig. 3, a top-level category would correspond, for example, to categories at the level of the categories 160, 170, 180. The user can
20 then select a category, for example, from a DDLB 560 labeled "Classification 2." When selected, the DDLB 560 will display categories that are in a level just below those in the DDLB 555. With reference to Fig. 3, these categories would correspond, for example, to categories at the level of the categories 190, 200,

210. The user interface 550 would provide additional levels of categories as needed to permit the user to select any category within the categorization scheme.

The user interface 550 can also provide the functionality that the selection
5 of a category in the DDLB 555 will narrow the displayed alternative categories when the DDLB 560 is selected. This permits the user to quickly work down the categorization scheme from a top level to lower levels without the need to sort through unrelated categories. After the user has selected a category using the user interface 550, the business process that is being performed can use the
10 business objects that are linked to the selected category.

In some circumstances, a run-time user may select any appropriate linked business objects to perform steps in a business process. In other examples, the run-time user may elect not to use any of the linked business objects, opting instead, for example, to use non-linked business objects, or to use no business
15 objects at all. After all steps that require categorization of the incoming message have been performed, then the run-time use of the coherent categorization ends.

In the foregoing example, business objects are linked to a selected category, and the business objects are used to perform a step in responding to the incoming message. The step may be performed once per incoming
20 message, or as many times as the run-time user provides an input command to perform that business process step. As such, user input determines which business process steps are performed in the presence of a particular incoming message. Whether multiple processes are performed or not, the categorization

is coherent if multiple business process steps are configured to be able to use business objects linked to a selected category.

One implementation of the business application 28 (Figs. 2A-2B) is an ERMS as shown in Fig 6. The ERMS may form part of an enterprise computing system 1 (Fig. 1A) to perform business processes other than those performed by the ERMS specific business application. As such, coherent categorization can be used in the enterprise computing system 1 to perform, for example, 1) a step in the ERMS business process, and then 2) a step in a different business process. In this connection, the other business process may be, but is not limited to, recording the interaction, performing service-related procedures, scheduling service orders, processing sales orders (e.g., 1-orders), data warehousing (e.g., SAP's Business Warehouse), and the like.

In Fig. 6, the result of a coherent categorization is first used by an ERMS business process 600 to respond to an incoming email message 610 by producing a response 612, and then to provide data to a different business process, namely a 1-order repository 632. When the ERMS business process 600 of this example receives the incoming email 610, a content analysis 614 is performed to analyze the contents of the incoming email 610. The analysis may incorporate, for example, a text mining engine (not shown) which provides text to be categorized to a categorization scheme stored in a categorization scheme repository 618. The result of the content analysis step 614 is a suggested category 615.

The suggested category is automatically suggested to a user in a categorization step 616. The categorization step 616 corresponds to the manual selection described with reference to Fig. 2B and Figs. 5A-5B. Nevertheless, in this manual embodiment, the user can choose whether to accept the suggested category 615, or to choose another category as the selected category 620.

The selected category 620 determines which API 622 is used to display the linked business objects. The API 622 defines, for example, the inheritance rules for displaying business objects. Inheritance rules may optionally be used to cause the display of business objects that are directly and/or indirectly linked to the selected category. For example, the inheritance rules may be configured to cause the display of all objects that are linked to the children of the selected category in addition to the objects directly linked to the selected category. In addition, the inheritance rules may optionally be configured to display business objects linked to parent categories of the selected category. The API 622 is typically configured when the software is installed in the enterprise computing system, and may be modified through maintenance. Accordingly, the API 622 can display business objects linked to parents and/or children of the selected category 620, in addition to the business objects in the set of linked business objects 624 that are directly linked to the selected category 620. The linked business objects 624, which corresponds to the linked business objects 44 in Fig. 2A, include experts 46, quick solutions 48, and/or response templates 50.

The linked business objects 624 represent stored information that is relevant to performing the ERMS business process 600, and specifically to

responding to the incoming email 610. For example, the experts 46 may identify a business partner who has special expertise that relates to the content of the incoming email 610. The quick solutions 48 may include documents that address the customer's questions in the email. In addition, the response templates 50
5 may provide the text of a reply email message so that the agent receives a prepared draft of a reply message.

Using these linked business objects 624, an agent can use an email editor 626 to finalize the response 612. Optionally, the agent may use other viewsets 628 to perform other steps in finalizing the response 612. For example, the
10 agent may use one of the other viewsets 628 to attach a document that is one of the quick solutions 48 in the linked business objects 624. The agent may also involve a subject matter expert in the response 612 by using an expert 46 in the linked business objects 624 to contact the subject matter expert.

In the final step of the ERMS business process 600, the agent ends the
15 contact 630 by, for example, sending the response 612 in the form of an email. Additional processes may be initiated as the contact is ended at 630. In this example, the 1-order repository 632 may record information about the just completed ERMS business process 600 for later uses. In other implementations, information about the transaction may be passed to other business processes
20 within the enterprise system 1 for purposes such as, for example, reporting, monitoring, quality control, and the like.

The just described exemplary ERMS business process 600 may include a number of business process steps that, when performed together, constitute a

system for responding to customer emails, and particularly business processes that are capable of supporting a large volume of interactions. Such business processes provide capabilities to interact with customers by e-mail, telephone, mail, facsimile, internet-based chat, or other forms of customer communication.

5 Such business processes may be manual, partially automated, or fully automated. Business processes that include automation generally use computers, which, in some implementations, take the form of enterprise computing systems that integrate and perform multiple business processes.

10 In this implementation, the content analysis step 614 involves selecting a category based upon the content of the incoming email 610. The content of the email 610 may be first be analyzed by, for example, a text-mining engine. In implementations, the content analysis step 614 may include identifying key words in the header or body, for example, of the incoming email 610. Key words may include words, phrases, symbols, and the like, that are relevant to performing the categorization. With reference to Fig. 3, categorizing the email 610 involves
15 selecting appropriate categories 110 under the appropriate categorization scheme 125 and based on the analyzed content of the e-mail 610.

As will be shown below (in Figs. 7-9), the computing system displays the business objects that are linked to the selected category 230. This display is
20 customized, as described above, using the categorization scheme objects API 622. The display of the linked business objects 624 allows the user to efficiently identify likely responses to the incoming email 610.

The linked business objects 624 that are displayed can be of at least three types. One type is an expert 46. Experts provide contacts and referrals to human resources who can provide knowledge and support that relates to the selected category 620. Referral of a request in an incoming email 610 to one or more experts 46 may constitute part of preparing the response 612. An expert may be, for example, a business partner (e.g., an independent contractor) who has a business relationship with the enterprise, although not necessarily an employee relationship. A second type of linked business object 624 is a quick solution 48. Quick solutions 48 refer to stored business objects that contain information responsive to the incoming email 610. Quick solutions 48 include documents that directly contain the responsive information, as well as pointers to other sources of such direct information, such as, for example, internet hyperlinks, website addresses, and uniform resource locators (URLs). A third type of a linked business object 624 is a response template 50 that may be incorporated into the email editor 626 for the purpose of providing the agent pre-formatted, predefined content for an email. These response templates save the agent time in drafting the content of a response to each incoming email 610, thereby promoting the efficient performance of the ERMS business process 600. Both quick solutions 48 and response templates 50 may be stored in a knowledge base or other information storage container (e.g., the stored information repository 8 of Fig. 1A) that may be accessed during run-time by business processes that use categorization schemes.

In the step of using the email editor 626 to finalize the response 612, the agent can review and edit the email. In addition, the user may also identify and attach to the email information, such as a quick solution 48 (e.g. documents or links to internet-based resources). Although the described implementation refers to preparing a response in the form of a reply email to the customer, other implementations may be used. For example, if an email is prepared, the email may be addressed to the customer who initiated the incoming email 610, or to an expert 46, or to both. However, the response 612 need not be in email form. By way of example, the response 612 may be in the form of a return phone call, facsimile, letter, or other action that may be internal or external to the enterprise system 1. If the incoming email 610 is a purchase order, for example, the response 612 may comprise an internally-generated sales order (via the 1-order repository 632) that ultimately results in the response 612 taking the form of a delivery of goods or services to the customer.

Depending upon the specific business process step that is being performed, the agent could also use the other viewsets 628 to finalize the response 612. The other viewsets 270 may be displayed as a part of a graphical user interface (GUI), as will be shown in Figs. 7-9. Example viewsets 628 include the following: e-mail editor, interaction log, attachment list, standard response query, value help selection query, standard response detail, knowledge search, search criteria, search results, and cart.

In implementations that are computer-based, portions of the business process steps to prepare the response 612 to the incoming email 610 may be

automated. For example, the categorization scheme repository 618 may be stored in a memory location, such as a disk drive, random access memory (RAM), or other equivalent media for storing information in a computer system.

In the end contact step at 630, for example, the results at the conclusion of the

5 ERMS business process 600 may be stored in a memory location, such as in a 1-order repository 632, for subsequent use. In the categorization step 616, as a further example, the process of categorizing is automated by using a programmed processor to rapidly execute a series of pre-programmed decisions to navigate a categorization scheme for the purpose of identifying which
10 predetermined categories are most relevant to performing the business process steps of responding to the incoming email 610.

In Figs. 7-9, a series of screen shots illustrates what an agent sees in the run-time environment 3 when executing the ERMS business process 600 of Fig. 6. In particular, the screen shots show an exemplary run-time graphical user
15 interface (GUI) by which an agent could achieve improved productivity by using coherent categorization to perform various steps in the ERMS business process 600.

In Fig. 7A, categorization of an incoming e-mail is used to automatically suggest e-mail response templates 50. A GUI 700 includes an e-mail editor
20 viewset 710 that includes text 712 from an incoming e-mail message that has already been received. Associated with the e-mail is the sender and recipient e-mail address information in an e-mail header viewset 714. Below the e-mail header viewset is an attachment viewset 716. When a response e-mail is

completed and submitted, the contents of the e-mail editor viewset 710, including the original text 712 and any text added by the agent, are e-mailed, along with any attachments identified in the attachment viewset 716, to the recipient in the e-mail header in viewset 714. The GUI 700 further includes an interaction record viewset 718 for monitoring and storing information about the reason for the interaction (see the interaction record business process step 120 of Fig. 3).

In this example, the agent has first entered information into the interaction record viewset 718 based upon the agent's analysis of the text 712 of the incoming message. The agent has specified that the reason for the e-mail relates to directions, that the priority of the interaction is medium, and that the e-mail may be described as relating to directions to LEGOLAND®. As one step of the ERMS business process, the information entered into the interaction record viewset 718 may be stored within the enterprise system 1 for later use.

The information that the agent has entered into the interaction viewset 718 provides the basis for performing a categorization using a categorization scheme. Given the above-entered information, and with reference to Fig. 3, the interaction record business process step 120 initiates a categorization through link 130 of the interaction reason categorization scheme 135. Moreover, because the reason relates to directions, the categorization traverses through the link 155 to the LEGOLAND® category 160, and from there, traverses through the link 205 to the driving directions category 210. Accordingly, the selected driving direction category 210 is linked by the link 235 to the set of linked business objects 240. The linked business objects 240, being linked to the selected category 210, are

used to perform the interaction record business process step 120. Because the categorization is coherent, the same linked business objects 240 may be used to perform other subsequent steps in the ERMS business process.

In Fig. 7B, the agent has initiated the step of creating the response email by selecting the drop down list box (DDL B) 730 in the email viewset editor 710. Having previously filtered out all business objects that are not linked to the selected driving directions category 210, the e-mail editor viewset 710 further filters out all business objects that are not response templates 50. A drop-down list box menu 730 displays four response template titles that are in the response templates 50 within the set of linked business objects 240. In this example, the agent can select from the four LEGOLAND® locations, namely Billund, California, Deutschland, and Windsor. According to the text 712 of the incoming message, the agent selects the appropriate response template that provides directions to LEGOLAND® California.

In this example, an analysis of the content of the email has identified that the incoming email request relates to driving directions. In response, the DDL B 730 displays a list of suggested standard responses that are linked to the selected driving directions category 210. The suggested responses include the response templates 50 from the linked set of business objects 240. As such, the suggested responses displayed in the DDL B 730 derive from a categorization based on the text 712 of the incoming email.

In Fig. 7C, the agent has selected an appropriate one of the suggested response templates 50. The text of the selected response template 50 has been

automatically entered into the e-mail editor viewset 712. With reference to Fig. 6, all that remains for the agent to do to finalize the ERMS business process 600 is to end the contact 630 and to submit the response 612. This example illustrates how business objects that are linked to a selected category may be used to perform a business process step, namely, the step of inserting an email response template into a response email.

Although, in the foregoing example, the agent selected one of the suggested response templates 50, the agent could have made other choices. For example, the agent could have selected the "More Responses..." from the DDLB 730 to display other business objects that are not linked to the selected driving directions category 210. Alternatively, the agent could have selected more than one of the response templates 50 for inclusion in the reply email.

In Figs. 8A-8F, a standard response template for driving directions to LEGOLAND® in California is processed in a different way than the example illustrated in Figs. 7A-7C. In Fig. 8A, instead of analyzing the text 712 of the incoming email and then entering information about the e-mail into the interaction record viewset 718, the agent first selects the DDLB 730 to manually select a category by navigating through a hierarchical categorization scheme. In this case, the agent selects the alternative "more responses.." in the DDLB 730 instead of any of the standard responses that are listed by default (not as the result of a categorization) in the DDLB 730.

In Fig. 8B, a "more responses" search viewset 810 is displayed in the GUI 700. Here, the agent selects the interaction reason field 812 to review the details

of available interaction reasons. With reference to Fig. 3, the agent will be able to review and select from among available categories within the interaction reason categorization scheme 135.

In Fig. 8C, a number of categories are listed with indications of hierarchical relationships. For example, three categories at a first level within a hierarchy correspond to the categories in Fig. 3 of LEGOLAND® 160, LEGO® CLUB 170, and LEGO® PRODUCTS 180. Under the LEGOLAND® category 160 are displayed the child categories of entry fee 190, events 200, and driving directions 210. Based upon the agent's analysis of the content of the incoming e-mail message, the agent has selected the driving directions category 210.

In Fig. 8D, four response templates 50 linked to the selected driving directions category 210 are displayed in a results viewset 820. Based upon the agent's analysis of the contents of the incoming email, the agent has selected the most appropriate response template 50, namely the directions to LEGOLAND® in California.

In Fig. 8E, the standard response detail viewset 830 displays the selected response template for the agent to review. The agent has selected the "insert" button 832 to insert this response template into the reply e-mail.

In Fig. 8F, the agent can review the reply email in the email editor viewset 710. The reply email 840 now includes both the text 712 of the incoming message and the selected response template 50. Having manually made the categorization selections as described above, the interaction record business process step 120 has been automatically performed using the selected driving

directions category 210. In the interaction record viewset 718, the reason and description have been automatically filled-in based upon the categorization.

The ERMS business process step 125 of replying to an e-mail has been performed. The agent has manually categorized the content of the incoming
5 email using the interaction reason categorization scheme 135. After the agent selected the driving direction category 210, a response template 50 linked to that selected category 210 was included in the response. In addition, the selected driving directions category 210 was also used to perform the interaction record business process step 120. Accordingly, the interaction record categorization
10 scheme 135 is coherent in this example because the selected category 210 was used to perform both the ERMS business process step 125 and the interaction record business process step 120.

In Figs. 9A-9D, a coherent categorization scheme is illustrated by an example in which a category selected for the interaction business record process
15 120 is also used by the ERMS business process 125 to identify both a response template 50 and a quick solution 48.

In Fig. 9A, the agent has entered information about the incoming email message 912 into the interaction record viewset 718. The information entered by the agent is based upon the agent's analysis of the content of the incoming e-
20 mail message 912.

In Fig. 9B, the GUI 700 responds by displaying an alert message 920 to indicate that automatically proposed solutions are available. The alert message 920 indicates to the agent that the information entered into the interaction record

viewset 718 has been categorized, and a category having attributes that match the entered information has been selected. Being alerted to this message, the agent looks for the proposed solutions by, for example, selecting a hyperlink associated with the alert message 920.

5 With reference to Fig. 3, the information entered into the interaction record viewset 718 in this example corresponds to the interaction record business process step 120, the interaction reason categorization scheme 135, the Lego® products category 180 and the building instructions category 220. As such, selecting the alert message 920 leads the agent to a viewset that displays
10 suggested business objects that are in the set of business objects 250, which is linked by the link 245 to the chosen building instructions category 220.

 In Fig. 9C, a knowledge search viewset 930 allows the agent to perform free-text searches for business objects in, for example, the stored information repository 8 (Fig. 1A). With reference to Fig. 6, the knowledge search viewset
15 arises in the viewsets 628. The knowledge search viewset 930 has a number of sub-viewsets, including a search criteria area 932 for inputting search terms and queries, a search results area 934 for selecting business objects retrieved by the search, and a cart area 936 for displaying selected business objects for later attachment to the reply email. In this example, the reason and the interaction
20 information record information from the interaction record viewset 718 (Figs. 9A-9B) automatically appear in the search terms dialog box in the search criteria area 932.

In the search results area 934, a list of search results is displayed. In this example, two search results are displayed, each of which corresponds to a quick solution 48 document. With reference to Fig. 3, the proposed quick solutions 48 are in the set of linked business objects 250 because the building instructions category 220 is selected. The displayed titles in the list may be in the form of hyperlinks. In some implementations, selecting a title in the search results area 934 causes the quick solution to be included in the cart area 936. In this example, the agent has selected one of the two quick solution 48 documents in the search results area 934, and the selected document is automatically displayed in the cart area 936.

In Fig. 9D, an attachments viewset 942 includes the quick solutions 48 that were placed in the cart area 936 (Fig. 9C). Not only has the selected quick solution 48, namely, the “Lego® Krikori Nui Building Instructions” document, been included as an attachment to the e-mail, but the DDLB 730 has also been automatically populated with a corresponding response template 50. The agent has selected the suggested response template 50 in the DDLB 730. Accordingly, the text 940 associated with the corresponding response template 50 has been inserted into the e-mail adjacent to the original text 912.

In the foregoing example, two business process steps have been performed using business objects linked to a single selected category. The selected building instructions category 220, which was initially selected during the performance of the interaction record business process step 120, has been used in the ERMS business process to perform the step of attaching a suggested

quick solution 48 to the reply e-mail, and to perform the step of inserting a suggested response template 50 into the reply e-mail.

With reference to Fig. 3, the interaction record business process step 120 was performed in response to the agent's entry of content analysis information into the interaction record viewset 718. This triggered a categorization of the entered information using the interaction reason categorization scheme 135. The selected building instructions category 220 is linked to the set of business objects 250. The set of business objects 250 was used to perform two business process steps. First, the quick solutions 48 of the set of linked business objects 250 were used to select a quick solution document to attach to the reply email. Second, the response templates 50 were used in the step of inserting response templates into the reply email. Accordingly, business objects that are linked to a selected category are used to perform multiple business process steps in the presence of an incoming message. As such, the example illustrates how a coherent categorization scheme can be used in the run-time environment 3 to help the agent prepare an e-mail with very little effort and with very little investment of time.

The foregoing examples have illustrates how quick solutions 48 and response templates 50 are types of linked business objects 44 that may be used to perform a business process step. As has been described above, experts 46 are another type of business object that can be linked to a selected category. In an ERMS business process, for example, using an expert 46 involves routing an electronic message to notify and to inform a human expert about the incoming

message. Each human expert has the capability to respond to certain categories of incoming messages. The capability of each human expert determines which categories are linked to each expert 46. Because experts that can provide high quality responses are limited resources, and because retaining experts can be
5 costly to an enterprise, the efficient allocation of the time of experts is an important factor in enterprise system cost and quality. Accordingly, the ability to refer only appropriate incoming messages to experts, or routing incoming messages to the appropriate experts, is important.

A computer system can receive, process and route electronic messages,
10 and it can route incoming messages to human experts. For example, a server device may be operably connected to the Internet, through which it can receive emails and other electronic messages sent from one or more other Internet-connected system(s). The server device may include an email program in which received messages, such as an exemplary incoming electronic message (IEM),
15 are accessible. The IEM may be processed to determine what action(s), if any, to take in response to receiving it. For example, the processing may lead to the IEM being routed to one or more individual accounts in the email program. The individual accounts may be assigned to persons working within the enterprise (or organization). For example, each expert can be assigned one of the individual
20 accounts. Accordingly, human experts working within the organization can access their respective individual account using a client device that is connected to the server device through a network. In an actual implementation, there may be many thousands of messages, or even hundreds of thousands or more. As

such, the computer system can route incoming messages to experts in order to perform various steps in the ERMS business process.

Creating Categorization Schemes

In order to be able to perform the above-described coherent categorization, a suitable categorization scheme must first be defined. One convenient method of defining a coherent categorization scheme is to use a design-time graphical user interface (GUI) like that illustrated in Fig. 10. With a user interface 1000, a user can arrange categories within the categorization area 1001 to have hierarchical relationships within the categorization scheme. The categorization area 1001 displays the names of categorization schemes and categories in rows. The user can enter, modify, and display categories in the categorization area 1001.

Accordingly, the categorization area 1001 in the user interface 1000 serves as a tool to enter, modify, and display categorization schemes. As can be appreciated, the categorization area 1001 is used to define various links that structure the hierarchical relationships within the categorization scheme. With reference to Fig. 3, the categorization area 1001 is used to define the links 155, 165, 175 between the categorization scheme 135 and the categories 160, 170, 180. Furthermore, the categorization area 1001 is used to define the links 185, 195, 205 between the parent category 160 and the child categories 190, 200, 210. However, the categorization area 1001 in this example does not (by itself) define links between business process steps and categorization schemes, or

between categories and business objects. In this example, those links are defined in conjunction with the linking area 1002.

In the linking area 1002, a number of tabs are provided to display various fields related to a user-highlighted category in the categorization area 1001. In this example, the driving direction category 210 is the highlighted category in the categorization area 1001. In Fig. 10, the query viewset tab 1003 is selected. In this example, the user interface 1000 is used to define a query for the highlighted category. The defined query can be evaluated to determine if the content of an e-mail corresponds to that category.

In the query viewset tab 1003, two rows of query criteria are shown. Elements for defining a query may be entered into columnar fields defined in a first row 1004 and a second row 1006. In the first row 1004, a match column 1005 includes a leading "if" statement. In the second row 1006, the match column 1005 includes a user-selectable drop-down list box (DDLb) into which the user can select various conditional conjunctions such as, for example, "and," "or," and "nor." The conjunction provides the logical operation that connects queries in the rows 1004, 1006. If at run-time, for example, the row query 1004 evaluates as "true" and if the row query 1006 evaluates as "false," and if the conjunction 1005 in the row 1006 is "or," then the complete query will evaluate as "true." However, if the conjunction 1005 in the row 1006 is "and," then the complete query will evaluate as "false." If the complete query for a category evaluates as "true," then the content of the e-mail "corresponds" to that category. On the other

hand, if the complete query evaluates as “false,” then the content does not correspond to that category.

Columnar fields in each row define the row queries for rows 1004, 1006.

An attributes column 1007 provides a DDLB through which the user can identify

5 attributes that are to be evaluated using the query defined in that row. For example, if the query of an e-mail relates to information contained in both the subject line and the body of the email, each row query can evaluate the content of the subject line, the body, or both. In this example, the row 1004 will evaluate “subject and body,” while the row 1006 query evaluates only the “subject.”

10 An operator column 1008 provides a DDLB through which the user can define the relational operator to be used to evaluate the query in that row. For example, the operator column 1008 may include operators such as equality, inequality, greater than, less than, sounds like, or includes. A value column 1009 provides a field in each of rows 1004, 1006 into which the user can enter values

15 for each row query. If the attribute 1007 and the value 1009 in a row query have the relationship of the selected operator 1008, then that particular row will evaluate as “true.” If the attribute 1007 and the value 1009 do not have the relationship of the selected operator 1008 for a particular row, then that particular row will evaluate as “false.” Each row is connected to the previous row or to the

20 subsequent row through a logical match operator 1005, such as “and,” “or,” and “nor.” Although only two rows 1004, 1006 are shown in this example, other rows may be entered using the scroll keys 1011. A case column 1012 provides a check box which, when checked, makes the query in that row case sensitive.

In accordance with the foregoing discussion, the above-described user interface 1000 can be used in the design-time environment 2 to create categorization schemes for use in the run-time environment 3. An exemplary procedure for using the user interface 1000 to efficiently define a categorization scheme is presented in Fig. 11. The exemplary method will be described with reference to previous figures. Other implementations of the method may use different steps, or be performed in a different order to achieve similar results. The example of Fig. 11 merely illustrates one representative implementation of the invention. However, the user interface 1000 is capable of being used in many different orders and combinations of steps.

A flow chart 1100 of the design-time procedure to create a categorization scheme using the user interface 1000 is shown in Fig. 11. Starting at 1105, the design-time developer selects a business process tab (e.g., ERMS tab 1013 in Fig. 10). At 1115, the user enters a categorization scheme into the categorization area 1001. With the categorization scheme highlighted, the user links at 1120 the categorization scheme to one or more application areas, which are also called business process steps (not shown in Fig. 10). With reference to Fig. 3, step 1120 corresponds, for example, to defining the links 130, 145, or 150.

At 1125, the user selects a categorization scheme in the categorization area 1001. At 1130, the user enters a category under the selected categorization scheme. At 1135, the user enters information about the selected category. At 1140, the user defines a query under the query tab 1003 that, if evaluated as true in the run-time environment 3, causes the associated category to be selected. At

1145, the user links a business object to the selected category by, for example, identifying a document for attachment.

At 1150, if the user wishes to add another business object to the selected category, the user can repeat the step at 1145 by, for example, selecting the standard responses tab 1014 and adding or modifying response templates (which are a type of business object). If the user does not wish to add more business objects, then at 1155, the user can choose to add another category under the current categorization scheme by looping back to step 1130. If no more categories need to be added, then the categorization creation process ends at 1160.

The foregoing exemplary method illustrates the entry of categories and the linking of business objects to create a categorization scheme. As can be appreciated, the method can be adapted as needed to use the user interface 1000 to modify and to display the categorization scheme that has been entered.

It can further be appreciated that the method can be used to enter, modify, and display more than one categorization scheme. Moreover, the method can be adapted so that the user interface 1000 can be used to maintain existing categorization schemes over time. For example, the user interface 1000 can be used to adapt the categories, linked business objects, and linked application areas to meet changing demands of the business processes as they evolve in the run-time environment 3.

Other Implementations

In implementations, coherent categorization schemes may be used within an enterprise to yield further benefits. For example, when used on-line, the semantics embodied in a categorization scheme can foster a higher level of automation in performing business processes. In addition, when used off-line, an enterprise can analyze a semantic footprint of its daily business by monitoring and reviewing the number of times each aspect, category, link, and business object in the categorization scheme is used. This off-line review can provide an enterprise with insight into how it might improve its business processes.

Because the categorization in this off-line review is coherent, the semantic expressions represent relationships among business objects and multiple business processes. Thus, coherent categorization may be used both on-line and off-line to improve integration and coordination between business objects and business processes in an enterprise computing system.

In some implementations, a categorization may be coherent where multiple business process steps use business objects linked to more than one selected category. This situation may occur where the analyzed content of an incoming message results in the navigation of a categorization scheme such that more than one category has attributes that match the analyzed content. As such, coherent categorization schemes may be used in implementations in which a number of categories are selected, so long as each selected category has links to business objects that may be used to perform more than one business process associated with the categorization scheme.

In some implementations, business processes may have multiple business processes, and each business process may have multiple steps. A categorization scheme that is associated with multiple business process steps may be associated with business process steps within a single business process, or with steps that span multiple business processes. More than one categorization scheme in an enterprise computing system may be associated with multiple business processes or business process steps. For purposes of being associated with categorization schemes, this document treats business processes and business process steps as being generally interchangeable.

The particular labels used above to describe implementations of the business processes, categorization schemes, and categories and business objects are merely exemplary. Other labels may be used to represent different implementations. For example, the categorization scheme labeled "interaction reason" may have a category labeled "hard disk crash." In this example, business objects that may be linked to the "hard disk crash" category are, for example, quick solution documents that describe how to recover from a hard disk crash, and e-mail response templates that are used to respond to severe hardware problems.

In implementations, at least some of the categories in a categorization scheme are linked to business objects. However, business objects are not limited to being linked to terminal categories. Accordingly, any category in a categorization scheme may be linked to one or more business objects.

In various implementations, business objects may be referred to as knowledge entities, business entities, and the like. As used in this document, these terms generally refer to information contained in an enterprise computing system, and, as such, may be stored using various known techniques for storing information in a computing system. In addition to data and program code, business objects may include links, such as Internet hyperlinks, to information stored outside of, but accessible by, the enterprise computing system. Such information may be stored or transmitted in digital or analog format. Business objects may also include information stored as a computer program product tangibly embodied in an information carrier. As such, the computer program product may contain instructions that, when executed, cause a processor to perform operations according to the implementations described in this document. The information may be stored in an arbitrary location, or in a structured knowledge base, database, information repository, or other equivalent structure for storing information.

With respect to business objects of the type "expert," the "stored" information may be within the knowledge of a human expert who may be referred to in responding to an incoming message. Typically, an expert has more capability to address certain categories of incoming messages than a general call center agent. "Experts" (also referred to as business partners) may refer to one or more individuals who may be employees or contractors, and who may be on-site or off-site relative to the physical enterprise computing system. Accordingly, references in this document to an expert business object refer to identifying

information, such as contact information, stored in the enterprise computing system. As such, a stored expert-type business object may provide a name, phone number, address, email address, website address, hyperlink, or other known methods for communicating with an expert who is linked to a selected
5 category.

Although the examples discussed in this document have focused primarily on business processes that handle inbound and outbound information in the form of email, the coherent categorization schemes may be used with other forms and combinations of inbound and outbound information. Such other forms of
10 information may include, for example, internet-based chat, data transmitted over a network, voice over telephone, voice over internet protocol (VoIP), facsimile, and communications for the visually and/or hearing-impaired (e.g., TTY), and the like. Furthermore, received information may be in one form while response information may be in a different form, and either may be in a combination of
15 forms. In addition, inbound and outbound information may incorporate data that represents text, graphics, video, audio, and other forms of data. The interaction may or may not be performed in real time.

One example of enterprise computing software that can benefit from creating run-time modules that use coherent categorization is an email response
20 management system (ERMS). In Fig. 1A, an enterprise computing system 1 may be used to design and run (i.e., execute) an ERMS. Designing and executing an ERMS involves moving, processing, and storing information in the system 1. The ERMS may be defined to perform a particular business process, or business

process step, in the design-time environment 2. After the design has been completed, the ERMS operates in the run-time environment 3.

Furthermore, the use of coherent categorization schemes may have applicability to processes other than the exemplary ERMS business process. For example, Internet-based self-service portals, or any computer-implemented (semi-automated) business applications in which multiple business process steps could benefit from coherency.

The invention can be implemented with digital electronic circuitry, or with computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors.

Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. The essential elements of a computer are a processor for executing instructions and a memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices

5 include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as

10 internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

The invention has been described in terms of particular implementations. Other implementations are within the scope of the following claims.